

## DATA ITEM DESCRIPTION

**Title:** Software Resources Data Reporting: Final Developer Report and Data Dictionary

**Number:** DI-MGMT-81740A

**Approval Date:** 20110518

**AMSC Number:** D9197

**Limitation:**

**DTIC Number:**

**GIDEP Applicable:**

**Preparing Activity:** CAPE

**Applicable Forms:** Software Resources Data Reporting: Final Developer Report  
(Sample Format 3)

**Use/Relationship:** This Data Item Description (DID) contains information about the sample format, content, and intended use for the data deliverable resulting from the task in the statement of work. This data deliverable consists of two parts. The first part, the Final Developer Report, is used to obtain the actual (as-built) characteristics of a software product and its development process. The second part is the Software Resources Data Reporting (SRDR) Data Dictionary, which defines each of the data elements within the Software Resources Data (SRD) report and describes the methods and rules used to perform the data measurement. Every submission of this SRD report shall contain both the SRDR Final Developer Report and an associated SRDR Data Dictionary.

The SRD report is not a management or software metrics report. It is not intended for tracking progress of the development during contract execution, nor is it intended to collect financial information. It does, however, collect the person-hours expended during software development.

The intent of the SRDR process is to collect objective measurable data commonly used by industry and DoD cost analysts. These data are used to compile a repository of actual software product sizes, schedules, effort, and quality that Government analysts can draw upon to build credible size, cost, and schedule estimates of future software-intensive systems.

Information to be acquired through these data will include descriptive information about the product and developer and actual as-developed software product size, development schedule, peak staff, and direct labor hours incurred.

The contractor must provide an SRDR Data Dictionary that defines the data elements contained in the negotiated SRDR Final Developer Report. The definitions of the data items are negotiable but must include the following categories of data: Context, Project Description, Size, Effort, and Schedule. Optionally, data in the category of Quality may be provided.

The minimum level of detail to be reported in each SRDR submission shall be in accordance with the contract's Cost and Software Data Reporting (CSDR) Plan, DD Form 2794, as approved by the Office of the Secretary of Defense (OSD) Deputy Director, Cost Assessment (DDCA). Discrete reporting is required for each Work Breakdown Structure (WBS) element identified in Box 13 of the CSDR Plan.

An SRDR submission shall be prepared in a Microsoft Excel-compatible electronic file format. For submissions that require discrete reporting of multiple WBS elements, the data shall be prepared and integrated into one electronic file.

The SRDR Data Dictionary shall be prepared in a readable electronic (digital) file format such as Microsoft Excel or Microsoft Word (e.g., pdf files are not acceptable).

All required SRDR submissions must be submitted by report upload to the Defense Cost and Resource Center (DCARC) secure Web site using the CSDR Submit-Review System. Uploading requires the use of a DoD Common Access Card (CAC) or a DoD-approved External Certificate Authority (ECA) certificate. See the DCARC Web site for certificate instructions.

This DID supersedes DI-MGMT-81740.

### **Requirements:**

1. Reference documents. The applicable issue of the documents cited herein, including their approval dates and dates of any applicable amendments, notices, and revisions, shall be as cited in ASSIST at the time of the solicitation; or, for non-ASSIST documents, as stated herein.

#### 1.1. References.

- 1.1.1. DoD Instruction 5000.02, "Operation of the Defense Acquisition System," [current version], available at <http://www.dtic.mil/whs/directives/>. This instruction establishes mandatory policies for requiring SRD reports.
- 1.1.2. DoD 5000.04-M-1, "Cost and Software Data Reporting (CSDR) Manual," [current version], available at <http://www.dtic.mil/whs/directives/>. This manual prescribes procedures and instructions for stakeholders in the SRDR process.

2. Format. There is no prescribed data format for either the SRDR Final Developer Report or the SRDR Data Dictionary. The SRDR Final Developer Report shall be in a format agreed to by the contractor and the Government. Software Resources Data Report: Final Developer Report (Sample Format 3), available on the DCARC web site, serves as a starting point for developing a tailored report. All required SRDR submissions must be submitted by report upload to the DCARC's secure Web site using the CSDR Submit-Review System.

3. Content. The SRDR Final Developer Report shall contain actual, as-built software measurement data as described in the contractor's SRDR Data Dictionary. The data shall reflect scope relevant to the reporting event. SRDR submissions for contract complete event shall reflect the entire software development project. When the development project is divided into multiple product builds, each representing production level software delivered to the government, the submission should reflect each product build. SRDR submissions for completion of a product build shall reflect size, effort, and schedule of that product build. The SRD report shall contain mandatory data elements as outlined below. Data elements reported beyond those outlined in this DID shall be agreed upon by the Cost Working-group Integrated Product Team (CWIPT) and approved by the OSD DDCA.

#### 3.1. Report Context and Development Organization.

- 3.1.1. Security Classification. The top and bottom of every page shall be marked with the security classification of the report which typically will be

“Unclassified”. However, if the appropriate security classification based on the classification level of the data reported is classified, contact the DCARC for special processing instructions.

3.1.2. Major Program.

a. Name: Enter the name given to the Major Defense Acquisition Program (MDAP) or Major Automated Information Systems (MAIS) as shown in the DDCA approved CSDR plan.

b. Phase/Milestone. Enter the appropriate Phase/Milestone which is being reported: Pre-A (Material Solution Analysis Phase), A (Technology Development Phase), B (Engineering and Manufacturing Development Phase), C-LRIP (Production and Deployment Phase – Low-Rate Initial Production), C-FRP (Production and Deployment Phase – Full-Rate Production), or O&S (Operations and Support Phase).

3.1.3. Reporting Organization Type.

- i. For a prime or associate contractor, check “Prime/Associate Contractor.” (See the “Definitions” section of this DID.)
- ii. For a direct-reporting subcontractor, check “Direct-Reporting Subcontractor.” (See the “Definitions” section of this DID.)
- iii. For a Government organization, check “Government.”

3.1.4. Name/Address.

a. Reporting Organization. Enter the name and address (including ZIP code) of the reporting organization actually performing the work.

b. Division. Enter the reporting organization’s division name and address (including ZIP code) if different than the performing organization.

3.1.5. Approved Plan Number. Enter the Approved Plan Number from Item 9 of the current OSD DDCA-approved contract or subcontract CSDR Plan that authorized the collection of data for this report.

3.1.6. Customer (Direct-Reporting Subcontractor Use Only). Enter the name of the Prime Contractor for whom the work on subcontract is being performed.

3.1.7. Contract Type. If the data are reported for a contract, enter the contract type code for the contract for which data are being reported. If the data are in response to a solicitation in accordance with DFARS sections 234.7101, 252.234-7003, and 252.234-7004, and the contract type has not been determined yet, enter NA (for “not applicable”). The codes for the common acquisition contract types included in the Federal Acquisition Regulation (FAR) are listed in the table below. For Time and Material, Labor-Hour, Letter Contracts, Indefinite Delivery, Basic Ordering Agreements, and flexibly priced contracts, select the primary contract type against which the majority of the orders are placed. If the contract type is an “Other Contract (OC),” enter “OC” in (“Contract Type”) and in Section 3.1.20 (“Remarks”) followed by the complete name of the contract type in Section 3.1.20. If the contract includes multiple Contract Line Item Numbers (CLINs) of

varying contract types, enter “MC” and in Section 3.1.20 followed by a description of the contracting arrangement in Section 3.1.20.

#### Contract/Order Type Code

FAR Contract Types	Contract Type Code
Cost Reimbursement Contracts	
Cost Sharing	CS
Cost Plus Award Fee	CPAF
Cost Plus Fixed Fee	CPFF
Cost Plus Incentive Fee	CPIF
Cost Plus Incentive Fee (With Performance Incentives)	CPIF(PI)
Fixed Price Contracts	
Firm Fixed Price	FFP
Fixed Price Incentive, Firm Target	FPIF
Fixed Price Incentive, Successive Targets	FPIST
Fixed Price Incentive, Successive Targets (With Performance Incentive)	FPIST(PI)
Fixed Price Incentive Firm Target (With Performance Incentive)	FPIFT(PI)
Fixed Price Award Fee	FPAF
Fixed Price with Economic Price Adjustment	FP/EPA
Fixed Price with Prospective Price Redetermination	FP/PPR
Fixed Ceiling Price with Retroactive Price Redetermination	FCP/RPR
Firm Fixed Price, Level of Effort Term	FFP/LOET
Indefinite Delivery, Indefinite Quantity	IDIQ
Letter Contract and Undefined Contractual Action (UCA)	LC
Time and Materials	TM
Other Contracts	OC
Contracts with multiple Contract Types by Contract Line Item Numbers (CLINS)	MC

#### 3.1.8. WBS Element Code/WBS Reporting Element.

*i.* A separate DD SRDR Initial Developer Report must be completed for each WBS Reporting Element for which an “X” is marked in Item 13f (Column “SRDR Formats”) of the OSD DDCA-approved contract or subcontract CSDR Plan. Enter the corresponding WBS Element Code identically as presented in Item 11b of the OSD DDCA-approved contract or subcontract CSDR Plan. For those elements designated with an “X” in Item 13f of the OSD DDCA-approved contract or subcontract CSDR Plan that reflect no expected costs at completion, or are marked “N/A,” a separate DD Form 1921-1 is not required, but may be submitted.

*ii.* Enter the corresponding WBS Reporting Element name as presented in Item 12 (column “WBS Reporting Elements”) of the OSD DDCA-approved contract or subcontract CSDR Plan.

#### 3.1.9. Type Action.

a. Contract No. and b. Latest Modification: Enter the assigned prime contract number the prime contractor has with the Government customer, as well as the

number of the latest contract modification. This requirement is identical for both reporting contractors and reporting subcontractors.

c. Solicitation No.: If the data are in response to a solicitation in accordance with, DFARS sections 234.7101, 252.234-7003, and 252.234-7004, enter the solicitation number.

d. Name: Enter the common reference name for the prime contract.

e. Task Order/Delivery Order/Lot No.: If the contract contains a task order(s), delivery order(s), and/or lot number(s) being reported on for which the CSDR Plan has reporting requirements, enter each as “TO,” “DO,” or “Lot” followed by a blank space and the applicable number.

3.1.10. Period of Performance. Enter the start and end dates related to the contractual period of performance. Enter the appropriate numeric data for the year, month, and day. For example, December 31, 2004, would be shown as 20041231.

3.1.11. Appropriation.

i. Check the appropriate box(es) to indicate the type of appropriation—Research, Development, Test and Evaluation (RDT&E), Procurement, or Operation and Maintenance (O&M)—used to fund the entire contract or the particular contract line item being reported on. The appropriation type must agree with the type specified in Item 9c of the CSDR contract plan, DD Form 2794.

ii. If the data are in response to a solicitation in accordance with DFARS sections 234.7101, 252.234-7003, and 252.234-7004, leave blank, unless otherwise specified in the solicitation.

3.1.12. Submission Number. Enter the submission number for the report provided in Item 14a of the current OSD DDCA-approved contract or subcontract CSDR Plan.

3.1.13. Resubmission Number. A resubmission occurs if prior submission(s) for the submission event were officially rejected with a memo signed by the DCARC Director. Enter “0” (zero) for original submission. If the report is a resubmission, enter the resubmission number, starting with “1” for the first resubmission, “2” for the second resubmission, and so on.

3.1.14. Report As Of. Enter the appropriate numeric data for the year, month, and last day of the reporting period. For example, December 31, 2004, would be shown as 20041231. The report as of date must be consistent with Item 14d of the OSD DDCA-approved contract or subcontract CSDR Plan. For CSDR Plans that include event-driven milestones for reporting purposes, any changes in the event date require appropriate adjustments to the “As of date” reported in Item 3.1.14 and to the OSD DDCA-approved CSDR Plan. In these situations, the Government Program Office, in coordination with the Contractor, must submit a request for change in the event-driven date for reporting through the CSDR Submit-Review system for DCARC approval before the date reflected in the OSD DDCA-approved CSDR Plan can be changed.

3.1.15. Point of Contact. Enter the following information for the person to contact for answers to any questions about entries on the Format 3: Date Prepared; Last Name, First Name, and Middle Initial; Department; Telephone Number, including Area Code; and E-Mail Address. For Date Prepared, enter the appropriate numeric data for the year, month, and day. For example, December 31, 2004, would be shown as 20041231.

3.1.16. Development Organization. Enter the name of the company or organization responsible for development of the software product. The associated SRDR Data Dictionary must be used to explain the mapping of development organizations, software components, and SRD reports submitted.

3.1.17. Software Process Maturity. The SRD report shall report the characterization of the developer's software process maturity using a methodology such as the Software Engineering Institute (SEI) software Capability Maturity Model (CMM), the Capability Maturity Model Integration (CMMI)-SW, or an alternative equivalent rating. The reported software process maturity shall reflect the rating that the primary development organization has formally certified as of the date of the reporting event. Identify the name of the person that performed the assessment, the evaluator's affiliation, and the date of certification. If no formal certification has been conducted, leave these items blank. If a single submission is used to represent the work of multiple organizations, enter the level of the organization that will be expending the most effort on the development project (not necessarily the prime contractor) and note this in the associated SRDR Data Dictionary. If the Government has accepted an alternative assessment mechanism, such as the Air Force's Software Development Capability Evaluation (SDCE) or ISO-15504, enter those results and explain the meaning of the assessment in the SRDR Data Dictionary.

3.1.18. Precedents. List up to five analogous systems developed by the same software organization or development team.

3.1.19. SRDR Data Dictionary Filename. If the SRDR Data Dictionary is stored as a separate electronic file from the SRD report, provide the filename of the SRDR Data Dictionary file. Provide the date the associated SRDR Data Dictionary was last revised.

3.1.20. Comments. Provide any comments about report context and development organization. Include more detailed explanations in the associated SRDR Data Dictionary.

### 3.2. Product and Development Description.

3.2.1. Functional Description. For each element reported, provide a brief description of its function. What is it? What does it do?

3.2.2. Software Development Characterization. In general language, provide a brief description for each element reported that characterizes the software development work undertaken on that element. Examples might include completely new from-scratch development, rehosting of software to different



processor/operating system, reengineering of legacy code into open architecture, translation of legacy code from Ada to C, and so on.

3.2.3. Application Type. Identify at least one application type (i.e., the end-user mission) developed using one or more domain names from those listed in Figure 1 (after the text in this DID). A minimum of one primary application type shall be identified, but any number of application types may be listed. If none of the examples in Figure 1 are appropriate, enter a phrase to describe the application type and define it in the associated SRDR Data Dictionary. When internal development efforts within a program are large and independent, respondents may choose to report each using a separate SRD report instead of as various application types within a single report. For every application type reported provide:

3.2.3.1. Primary and Secondary Programming Language. Enter the primary and secondary computer language in which most of the development was conducted. This can be a compiled language, such as FORTRAN, Ada, or C, an interpreted language such as Basic, or a graphical or model-based language, such as Rhapsody/UML or Simulink. Use the amount of effort spent in development to determine the primary language rather than the amount of function delivered. Explain any interpretation of this item in the associated SRDR Data Dictionary.

3.2.3.2. Percent of Overall Product Size. Enter the overall approximate percentage (up to 100%) of the product size that is of this application type. If relevant and appropriate, the contractor can include the integrated Commercial Off-the-Shelf (COTS)/Government Off-the-Shelf (GOTS) packages in this calculation. If so, an explanation must be placed in the SRDR Data Dictionary.

3.2.3.3. Actual Development Process. Enter the name of the development process followed for the development of the system. Do not indicate a software architecture method (such as object-oriented development) or a development tool (such as Rational Rose), as these do not specify a process. Typical types of development processes adopted include waterfall, spiral, or Rapid Application Development (RAD). If the contractor uses an atypical internal process was used, provide a description of the development process in the SRDR Data Dictionary.

3.2.3.4. Software Development Method(s). Identify the software development method or methods used to design and develop the software product (e.g., Structured Analysis, Object Oriented, Vienna Development Method, etc.).

3.2.3.5. Upgrade or New Development? Indicate whether the primary development was new software or an upgrade. A software system is considered new either if no existing system currently performs its function or if the development completely replaces an existing system. A software system that replaces part of an existing system (such as the replacement of a database) should be considered an upgrade. An existing software system that was ported to a new platform or reengineered to execute as a Web or distributed

application (for example) would be considered an upgrade unless it was also completely redeveloped from scratch (new requirements, architecture, design, process, code, etc.).

3.2.3.6. Software Reuse. Identify by name and briefly describe software products reused from prior development efforts (e.g. source code, software designs, requirements documentation, etc.).

#### 3.2.4. COTS/GOTS Applications Used.

3.2.4.1. Name. List the names of the applications or products that constitute part of the final delivered product, whether they are COTS, GOTS, or open-source products. If a proprietary application or product that is not generally commercially available will be included, identify it here and include any necessary explanation in the associated SRDR Data Dictionary.

3.2.4.2. Integration Effort (Optional). If requested by the CWIPT, the SRD report shall contain the actual effort required to integrate each COTS/GOTS application identified in Section 3.2.4.1. “Effort” may be expressed in terms of staff-hours, new/modified glue code, or a qualitative assessment of effort required (i.e., low, medium, high, etc.). The SRDR Data Dictionary shall contain appropriate definitions of the integration effort metric chosen by the contractor.

#### 3.2.5. Staffing.

3.2.5.1. Peak Staff. For the element reported, enter the actual peak team size, measured in full-time equivalent (FTE) staff. Include only direct labor in this calculation unless otherwise explained in the associated SRDR Data Dictionary. The SRDR Data Dictionary shall include a definition of FTE that includes the hours per staff-month used to compute FTE.

3.2.5.2. Peak Staff Date. Enter the date when the actual peak staffing occurred.

3.2.5.3. Hours per Staff-Month. Enter the number of direct labor hours per staff-month. Indicate in the SRDR Data Dictionary whether the reported hours per staff-month reflect an accounting standard or a computation. If they reflect a computation, provide details on how the computation was performed.

3.2.6. Personnel Experience in Domain. Stratify the project staff domain experience by experience level and specify the percentage of project staff at each experience level identified. (Sample Format 3 identifies five levels: Very Highly Experienced (12 or more years), Highly Experienced (6 to 12 years), Nominally Experienced (3 to 6 years), Low Experience (1 to 3 years) and Inexperienced/Entry Level (less than a year). Provide a definition for each experience level (i.e., the number of years of experience) in the SRDR Data Dictionary. Also provide a definition of domain experience in the SRDR Data Dictionary (e.g., “Domain experience is defined as the number of years a project staff member has worked within a mission discipline such as real time fire control radar or missile guidance and tracking”). Additionally, the contractor is permitted



to tailor the type of experience reported to track to whatever type of experience is most applicable (e.g., Ada programming experience, total software development experience, etc.).

3.2.7. Comments. Provide any comments about the product and development description. Include more detailed explanations in the associated SRDR Data Dictionary.

### 3.3. Product Size Reporting.

3.3.1. Number of Software Requirements. Provide the actual number of software requirements. The method of counting actual number of requirements implemented by the development software must be the same as that used for counting estimated requirements (as reported in the SRDR Initial Development Report). Do not count requirements concerning external interfaces not under project control (see next item, "Total Requirements"). Alternative requirements counts based on Use Cases are also permitted. The SRDR Data Dictionary shall provide both a definition of what types of requirements are included in the count (i.e., functional, security, safety, other derived requirements, etc.) and the units (e.g., "shall," "sections," paragraphs, etc.) and counting methods used.

3.3.1.1. Total Requirements. Enter the actual number of total requirements satisfied by the developed software product at the completion of the increment or project. This count must be consistent with the total size of the delivered software (i.e., it must not solely focus on new development, but must reflect the entire software product).

3.3.1.2. New Requirements. Of the total actual number of requirements reported, the SRD report shall identify how many are new requirements.

3.3.2. Number of External Interface Requirements. Provide the number of external interface requirements, as specified below, not under project control that the developed system satisfies. External interfaces include interfaces to computer systems, databases, files, or hardware devices with which the developed system must interact but which are defined externally to the subject system. If the developed system interfaces with an external system in multiple ways (such as for reading data and also for writing data), then each unique requirement for interaction should be counted as an interface requirement. Provide the actual number of interface requirements handled by the developed software using the same counting method as was used in the estimating (initial) reports. Explain any details about the counting methods for external interface requirements in the SRDR Data Dictionary.

3.3.2.1. Total External Interface Requirements. Enter the actual number of total external interface requirements satisfied by the developed software product at the completion of the increment or project. This count must be consistent with the total size of the delivered software (i.e., it must not solely focus on new development, but must reflect the entire software product).

3.3.2.2. New External Interface Requirements. Of the total number of external interface requirements reported, the SRD report shall identify how many are new external interface requirements.

3.3.3. Requirements Volatility. Indicate the amount of requirements volatility encountered during development as a percentage of requirements that changed since the Software Requirements Review.

3.3.4. Software Size.

3.3.4.1. Delivered Size. The SRD report shall capture the delivered size of the product developed, not including any code that was needed to assist development but was not delivered (such as temporary stubs, test scaffoldings, or debug statements). Additionally, the code shall be partitioned (exhaustive with no overlaps) into appropriate development categories. A common set of software development categories is new, reused with modification, reused without modification, carry-over code, deleted code, and generated code.

3.3.4.1.1. Reused Code With Modification. When code is included that was reused with modification, the contractor shall provide, in the SRDR Data Dictionary, an assessment of the amount of redesign, recode, and retest required to implement the modified or reused code. This amount should be reported as a percentage of redesign (artifacts that had to be changed compared to the total), percentage of recode (code changed compared to the total) and percentage of retest (test artifacts that had to be rewritten or created compared to the total). To assist with reporting, the following factors should be considered and reported for each category:

Redesign

- Required an architectural design change
- Required a detailed design change
- Required reverse engineering
- Required modification to existing documentation
- Required revalidation of the new design

Recode

- Required code changes
- Required code reviews
- Required unit testing

Retest

- Required test plans to be rewritten
- Required test procedures to be identified and written
- Required new test reports
- Required test drivers and simulators to be rewritten
- Required integration testing
- Required formal demonstration testing

3.3.4.1.2. Reuse Code Without Modification. Code reused without modification is code that has no design or code modifications. However, there may be an amount of retest required. Percentage of retest should be

reported with the retest factors described above. Code reused without modification may be further partitioned into reuse from a previous increment of this project (that was previously reported on an SRD report associated with that prior increment) or reused from a source external to the project. In all cases, the partitioning used for reporting shall be customized to conform to the contractor's standard internal reporting of software development categories as long as the partitioning does not double count or omit any delivered software.

3.3.4.1.3. Carryover Code. Do not count the same code as new in more than one SRDR incremental report. Except for the first increment SRD report or the overall project SRD report at contract completion, an SRD report shall distinguish between code developed in previous increments that is carried forward into the current increment and code added as part of the effort on the current increment. Examples of such carried-forward code include code developed in Spiral 1 that is included in Spiral 2 or code that is developed for Version 3.0 software that is included in Version 3.1 software. Table 1 provides one possible example of reporting code from previous builds for the Initial Developer Reports associated with builds.

Table 1: Example of Reporting Carryover Code from Previous Builds

		<b>Contract Start</b>	<b>Build 1 Start</b>	<b>Build 2 Start</b>	<b>Build 3 Start</b>
New Code	Human Generated	3,500	1,000	0	2,500
	Auto Generated	3,000	0	500	2,500
External Reused	With Modification	20,500	5,000	15,000	500
	Without Modification	5,000	3,000	0	2,000
Carryover Code from Previous Build	With Modification	N/A	0	0	12,250
	Without Modification	N/A	0	9,000	12,250
Total Delivered Code		32,000	9,000	24,500	32,000

3.3.4.1.4. Deleted Code. The SRD report shall include the amount of delivered code that was created and subsequently deleted from the final delivered code

3.3.4.1.5. Auto-generated Code. If the developed software contains auto-generated source code, the SRD report shall include an auto-generated code sizing partition as part of the set of development categories.

3.3.4.1.6. Subcontractor-Developed Code. The categories of delivered code in the SRD report shall be further partitioned by responsible developer, for example: Prime Contractor Only and All Other Subcontractors. If the subcontractor-developed code cannot be further partitioned, then report only total delivered code for the subcontractors. If the delivered size of one or more subcontractors is unknown, annotate in the SRD report form and provide additional explanation in the SRDR Data Dictionary.

3.3.4.2. Counting Convention. Identify the counting convention used to count software size. The specific definition must be provided in the SRDR Data Dictionary. While Source Lines of Code (SLOC) is a prominent unit of software size, the SRD report shall reflect units of measure in use internally to the contractor. Alternative units, such as function points, are permissible units of measure to report so long as the contractor consistently reports this on both the Initial Developer Report and the Final Developer Report. Units of measure that reflect weighted sum normalization of size into equivalent units, such as Equivalent New Lines of Code, shall not be used as a primary sizing unit of measure in the SRD report. This information may be reported in the SRD report's supplemental information.

3.3.4.3. Size Reporting by Programming Language (Optional). The SRD report shall partition software size by programming language if requested by the CWIPT.

3.3.4.4. Standardized Code Counting (Optional). If requested by the CWIPT, the contractor shall use a publicly available and documented code counting tool, such as the University of Southern California Code Count tool, to obtain a set of standardized code counts that reflect logical size. These results shall be used to report software sizing in the SRD report.

3.3.5. Comments. Provide any comments about product size reporting. Include more detailed explanations in the associated SRDR Data Dictionary.

3.4. Resource and Schedule Reporting. The Final Developer Report shall contain actual schedules and actual total effort for each software development activity.

3.4.1. Effort. The units of measure for software development effort shall be reported in staff-hours. Effort shall be partitioned into discrete software development activities as defined by the contractor's standard software development process. Software effort reporting by development activity is still required when software development activities are not discretely identified in the CSDR Plan WBS. The following activities are taken from activity definitions used by commercial software development organizations and are intended as an example of partitioning software development effort. Additional software support activities are also shown.

- software requirements analysis,
- software architecture and detailed design,
- software coding and unit testing,
- software integration,
- software qualification testing,
- system/software integration,
- system/software qualification testing,
- software quality assurance,
- software configuration management,
- software program management, and
- other software support activities:
  - data,
  - software process improvement,
  - IV and V, and
  - problem resolution.

3.4.2. WBS Mapping. For each software development activity reported in the Final Developer Report, identify, from the contractor's OSD DDCA-approved CSDR plan, the contract WBS name(s) and WBS number(s) that capture that software development activity. Do not reference internal contractor cost account codes elements.

3.4.3. Subcontractor Development Effort. The effort data in the SRD report shall be separated into a minimum of two discrete categories and reported separately: Prime Contractor Only and All Other Subcontractors. The prime contractor shall report the subcontractor's actual effort, if available. If the subcontractor's actual effort data are not available, then the prime contractor shall estimate the subcontractor effort and clearly annotate that the sub-contractor effort reflects and estimate. If the reported subcontractor development effort cannot be partitioned by software development activity, then report only the total effort expended and provide a definition in the SRDR Data Dictionary that explains what software development activities are included in the subcontractor-reported development effort.

3.4.4. Schedule. For each software development activity reported, provide the actual start and end dates for that activity. Alternatively, month numbers, starting with month "1" at the time of Contract Award, can be used. If there were multiple start and stop dates for the same activity, as would be the case for iterative or spiral development, then report the earliest and latest end date for each activity, to the extent that is sensible for the approach used. If month numbers are used,

provide the date that is equivalent to month “1” either as a comment in the SRDR Initial Developer Report or in the SRDR Data Dictionary.

3.4.5. Comments. Provide any comments about resource and schedule reporting. Include more detailed explanations in the associated SRDR Data Dictionary.

3.5. Product Quality Reporting. Quality should be quantified operationally (through failure rate and defect discovery rate). However, other methods may be used if appropriately explained in the associated SRDR Data Dictionary.

#### 3.5.1. Defects

3.5.1.1. Number of Defects Discovered. Report the number of defects discovered during integration and qualification testing. If available, list the defect discovery counts by priority, e.g. 1, 2, 3, 4, 5. Provide a description of the priority levels if used. If an existing metrics process captures counts for Defects Discovered during other software development activities, identify where in the development process by contractor defined activities defect counting occurs. If counts are provided by activity or phase, identify if the count is cumulative.

3.5.1.2. Number of Defects Removed. Report the number of defects verified and required to be removed in order to pass qualification testing. In addition, report the number of defects verified for deferral and will not be removed until post deployment. If available, list the defect removal counts by priority. If an existing metrics process captures counts for Defects Removed during other software development activities, identify where in the development process by contractor defined activities defect counting occurs. If counts are provided by activity or phase, identify if the count is cumulative.

3.5.1.3. Comments. Provide any comments about the product quality. Include more detailed explanations in the associated SRDR Data Dictionary.

3.6. SRDR Data Dictionary. The SRDR Data Dictionary shall contain, at a minimum, the following information in addition to the specific requirements identified in Sections 3.1 through 3.6:

3.6.1. Experience Levels. Provide the contractor’s specific definition (i.e., the number of years of experience) for personnel experience levels reported in the SRD report.

3.6.2. Software Size Definitions. Provide the contractor’s specific internal rules used to count software code size. The Software Engineering Institute (SEI) technical report, “Software Size Measurement – A Framework for Counting Source Statements,” has sample checklists for physical and logical code counting; however, the contractor may use any checklist. The rules must address (a) what is counted (i.e., terminal semi-colons, non-comment non-blank physical lines, etc.), and (b) what is included in the logical size count (i.e., job control language, included files, comments, etc.).



3.6.3. Software Size Categories. For each software size category identified (i.e., New, Modified, Unmodified, etc.), provide the contractor's specific rules and/or tools used for classifying code into each category.

3.6.4. Peak Staffing. Provide a definition that describes what activities were included in peak staffing.

3.6.5. Requirements Count (Internal). Provide the contractor's specific rules and/or tools used to count requirements. The definition must also identify the source document used for tallying requirements (i.e., system/subsystem design description, software specification document, etc.).

3.6.6. Requirements Count (External). Provide the contractor's specific rules and/or tools used to count external interface requirements. The definition must also identify the source document used for tallying requirements (i.e., SV-6, ICD, etc.).

3.6.7. Requirements Volatility. Provide the contractor's internal definitions used for classifying requirements volatility.

3.6.8. Software Development Activities. Provide the contractor's internal definitions of labor categories and activities included in the SRD report's software activity. This definition should not focus on a textbook software engineering definition, but should rather focus on the natural manner in which the contractor describes the kinds of efforts that are included in the software development activity.

3.6.9. Product Quality Reporting. Provide the contractor's internal definitions for product quality metrics being reported and specific rules and/or tools used to count the metrics. If a metric is based on computations involving two or more metrics, clear definitions of all metrics used must be provided along with a description of the formula used to include numbers resulting in the computed metric reported.

3.6.10. Comments. Provide any additional information that would permit a DoD cost analyst to correctly interpret the contractor's data.

Figure 1. Application Types

<b><u>Warfare Mission Areas</u></b> Anti-air Warfare Antisubmarine Warfare Naval Antisurface Ship Warfare Amphibious Warfare Chemical Warfare Biological and Radiological Defense Land Warfare Special Warfare Strategic Warfare Tactical Air Warfare Electronic Warfare Strategic Defense Initiative	<b><u>Defensive Systems Functions</u></b> Hit Avoidance Signature Control/Suppression Reduction Armor, Infantry and Crew Protection EMP Hardening/Survivability from Nuclear Weapons Damage Control Chemical/Biological Defense Deterrence	<b><u>Miscellaneous Functions</u></b> Multi-Function Applications Robotics Human Factors/Human Engineering Artificial Intelligence/Adaptive Systems Basic Scientific Research/ University Interactions
<b><u>Mobility Mission Areas</u></b> Air Mobility Land Mobility Sea-Surface Mobility Undersea Mobility Space Mobility	<b><u>Mine Functions</u></b> Mine Mooring Mine Neutralization/ Destruction	<b><u>Supply/Support/Construction Functions</u></b> Material Distribution and Payload Handling/Supply Systems Training Field Services (Water, Food, Tents, etc.) Bridging/Obstacles Support and Auxiliary Equipment Habitability Environmental Effects Facility Construction
<b><u>Communications, Command and Control/Intelligence Mission Areas</u></b> Communications, Command and Control Intelligence, Including Reconnaissance	<b><u>C3I Functions</u></b> Information Management Communication Guidance/Navigation/Position Location Avionics/Vetronics/ Display Systems	<b><u>Management/Personnel Functions</u></b> RDT&E Management Acquisition Management Financial Management Medical/Casualty Care Performance Appraisal
<b><u>Mine and Obstacle Mission Areas</u></b> Land Mine/Obstacle/ Countermeasures Sea Mine/Countermine	<b><u>Electronic Warfare Functions</u></b> Electronic Countermeasures Jamming Deception Cryptography Electronic Counter Countermeasures Low Probability Electromagnetic Signal Measurement/Intelligence Jam Resistance	<b><u>Other Embedded Functional Areas</u></b> Avionics Audio Signal Processing and Enhancement Command and Control Command, Control and Information Command, Control, Communications and Information Command, Control, Communications, Computers and Information Digital Signal Processing Guidance and Control Image Processing and Enhancement Operational Flight Program Simulation Telemetry Target Seeking Embedded Trainer Software Embedded Weapon
<b><u>Mission and System Support Mission Areas</u></b> Logistics Manpower, Personnel and Training Mission/System Support	<b><u>Assessment/Analysis Functions</u></b> Simulation Weapons and Munitions Effects/Target Kill Assessment Vulnerability Analysis	<b><u>Other Software System Functions</u></b> Decision Support Financial, Accounting, Bookkeeping, Payroll, etc. Information System Management Information System Personnel, Human Resources, etc. Operating System Online Training or Education Software
<b><u>Weapon Systems Functions</u></b> Target Acquisition/ Search/Detect Threat Evaluation Target Tracking Weapon Assignment Fire Control Acquisition and Designation Launch Propulsion Control Flight Controls Conventional Munitions/ Weapons Directed Energy Weapons Hard Target Kill/Anti-Armor Fuzing Chemical Warfare (Offense)	<b><u>RDT&amp;E Functions</u></b> Energetic Materials Manufacturing Technology Electronics Other Than Electronics Materials Development Metals, Ceramics, Organics and Composites Electronics Test Equipment/Technology Structural Electronics Reliability Maintainability Structures, Including Design and Manufacture Missile Aircraft Hull Body/Chassis	

End of DI-MGMT-81740A